Enhancing the Accuracy of Network-based Intrusion Detection with Host-based Context

Holger Dreger[1], Christian Kreibich[2],
Vern Paxson[3], **Robin Sommer**[1]

[1]TU München
Germany

[2]University of Cambridge
United Kingdom

[3]ICSI / LBNL
Berkeley, CA, USA

DIMVA 2005

Motivation

- Network intrusion detection systems (NIDS)
  - Deployed at central network location
  - Suffer from ambiguities and performance problems
- Host intrusion detection systems (HIDS)
  - Deployed on individual hosts
  - Suffer from performance overhead and maintenance hassle
- We combine the two approaches
  - Focus on network-based detection
  - Hosts supply additional context
- Advantages
  - Centrally managed security policy
  - Enhanced accuracy
  - Low performance overhead on host

Outline

Outline

Observation

- Server application analyzes its input
  - Parses client input (e.g., login sessions)
  - Decides how to react (e.g., deny access)
  - Sends appropriate response
- NIDS analyzes all connections
  - Decodes protocols
  - Extracts semantic information (e.g., user name)
  - Performs detection (e.g., sensitive logins)
- If NIDS could “see” the host’s analysis, it could either
  - Replace its own analysis or
  - Verify its own analysis
- We enable host to send information to the NIDS

Applications (1)

- Comprehensive protocol analysis
  - Applications include full protocol decoders
  - Hosts can supply internal protocol state
- Anti-Evasion
  - Evasion attacks exploit ambiguities
  - Host can provide authoritative view
- Overcoming encryption
  - NIDS cannot decode encrypted connections
  - Host can supply unencrypted data

Applications (2)

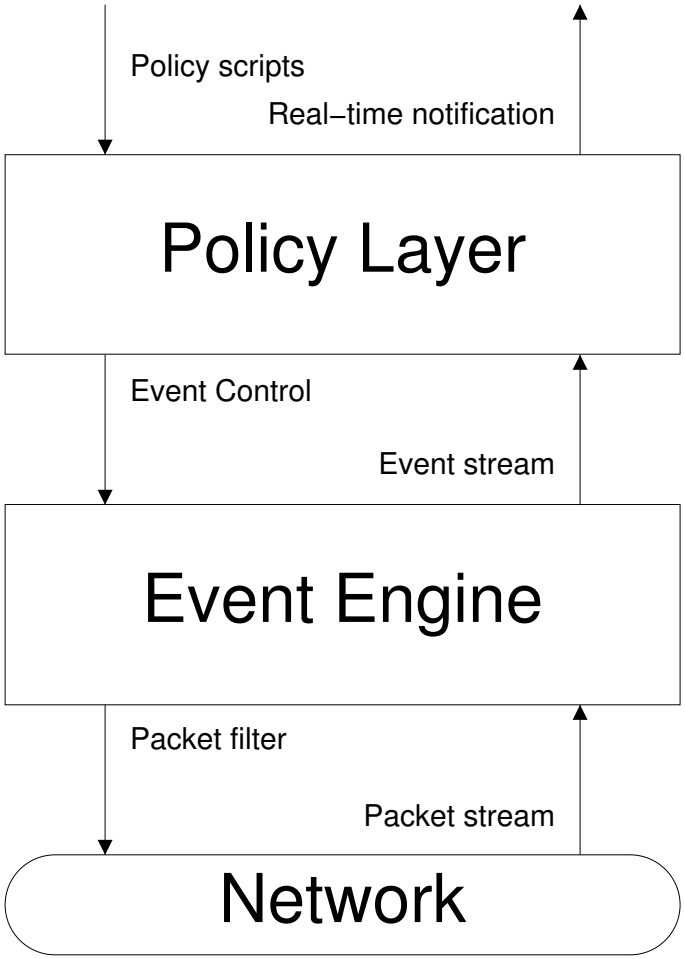
- Adaptive scrutiny
  - NIDS can increase depth of analysis for suspicious hosts
  - Host can signal suspicious activity
- NIDS hardening
  - NIDS needs to robustly decode protocols
  - Analysis mismatches may indicate a bug in the NIDS

Outline

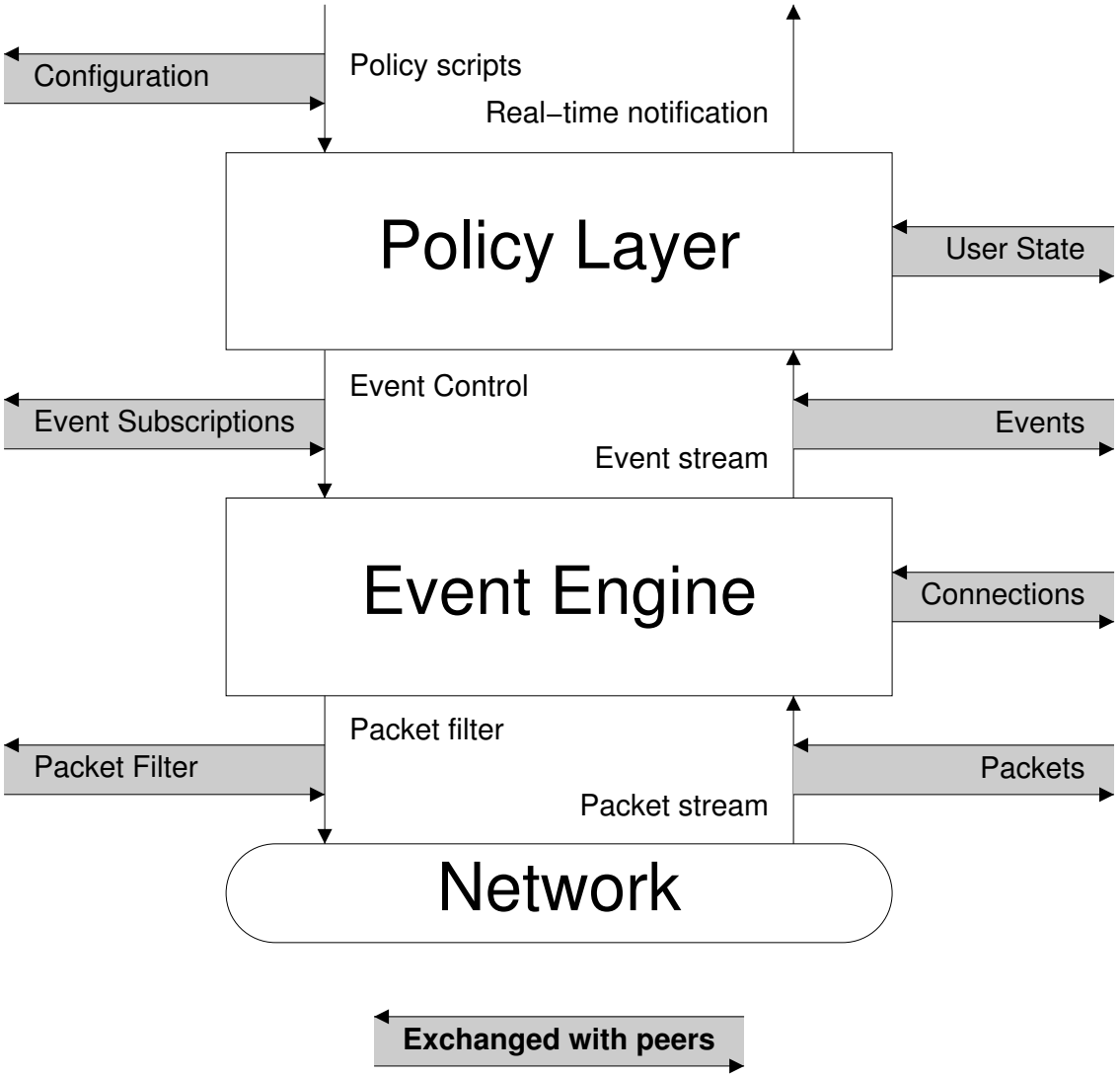
The Bro Network Intrusion Detection System

- Bro is powerful open-source NIDS
- Used in various high-performance networks
- Supports different approaches to intrusion detection
- Focuses on
  - Semantically high-level analysis
  - Efficiency
  - Extensibility
  - Robust operation
  - Separation of mechanism and policy

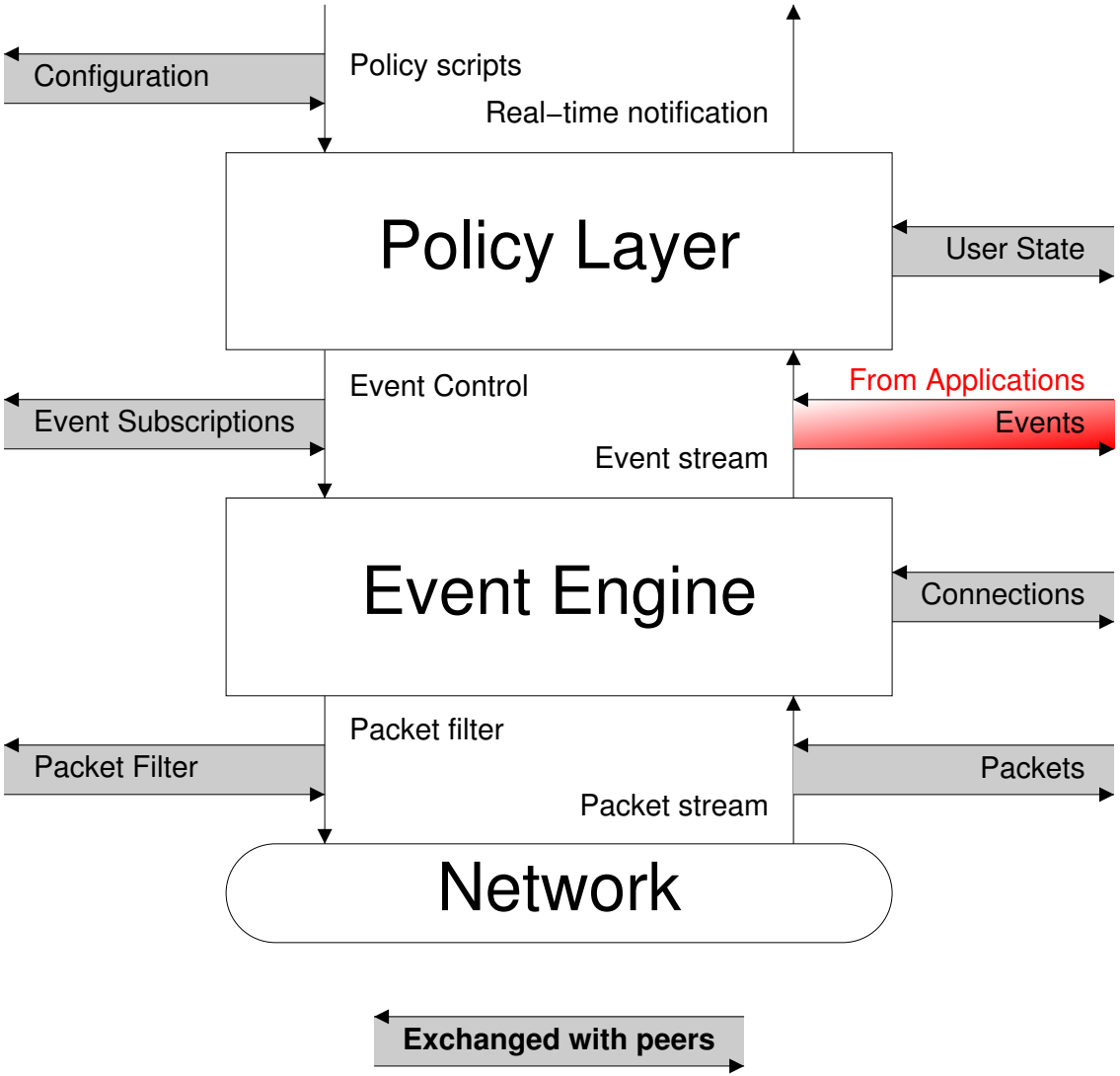
Bro's Architecture



Bro's Architecture



Bro's Architecture



Integrating Host-supplied Context Into Bro

- Applications send events to Bro
  - Events are abstractions of host activity
  - Events are policy neutral (like core events)
  - Events are inserted into stream of core events
- Bro maintains central policy
  - No individual configuration on hosts required
  - Bro's full toolbox is used to take decisions
- Application's overhead is low
  - Sending events is inexpensive
  - Instrumentation requires little effort
  - Client-side library is provided (*Broccoli*)
- Bro's overhead is low
  - Receiving events is inexpensive

Outline

Leveraging Web Server Context

- HTTP is most widely used application-layer protocol
- Requests are analyzed by two components
  - Network intrusion detection system
  - Web server
- Interfacing Web server to NIDS
  - Send client-requests to NIDS
  - Replace/supplement NIDS analysis
- Replacing NIDS's HTTP analysis provides
  - Off-load NIDS saves CPU cycles
  - Full request/reply analysis
  - Analysis of SSL sessions
- Supplementing NIDS's analysis provides
  - Detection of analysis differences (e.g., URLs)

Setups

- Implementation for Apache and Bro
  - Apache sends log-entries to Bro
  - Instrumentation done via module or log-pipe
- Installed Apache/Bro combo in three setups
  - Computer science's Web server of TUM
  - Work group's Web server at TUM
  - Test-bed setup for stress tests (libwhisker, Nikto)
- Implemented two kinds of analysis
  - Run Bro's standard analysis on requests/replies
  - Compare received requests with self-decoded
- Confirmed that our implementation works reliably
  - Reliably sees all requests (incl. SSL)
  - Detections works (incl. bi-directional signatures)

Differences between Apache and Bro

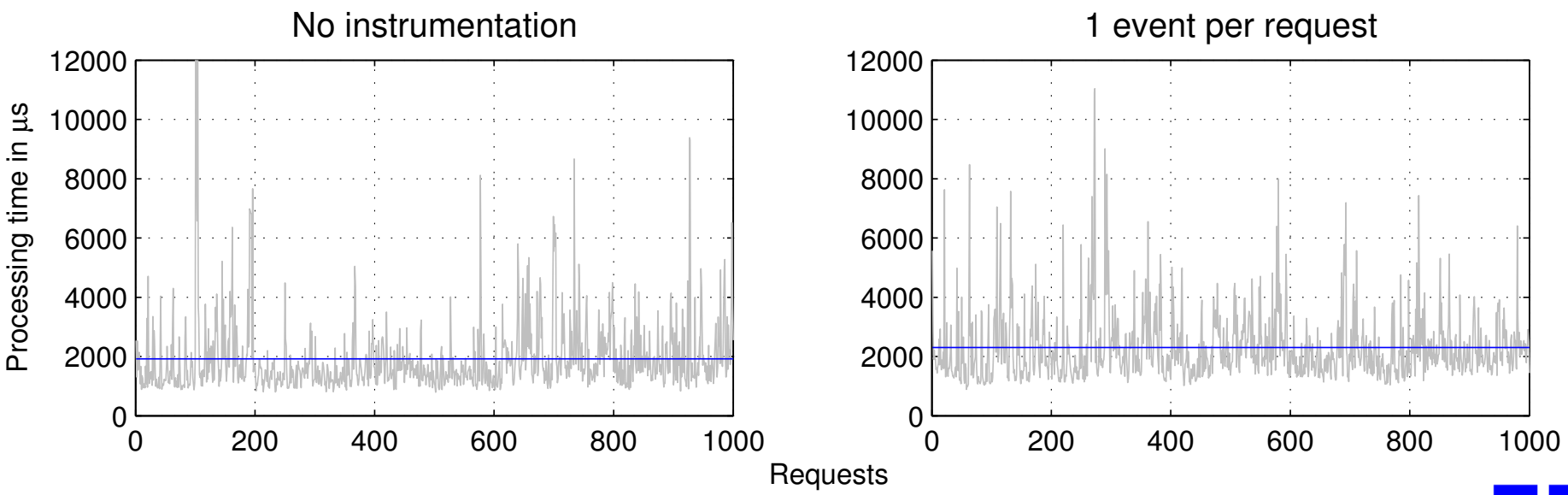
- Overall, Apache and Bro work well together
- Main differences between Apache and Bro
  - Apache's expansion and rewriting:
    `/foo/bar/` → `/foo/bar/index.html`
  - Different forms of URL canonicalization, e.g.,

	Request	Apache	Bro
(1)	`tmp/../i.html`	`i.html`	`tmp/../i.html`
(2)	`http://a.b/i.html`	`i.html`	`http://a.b/i.html`
(3)	`i%%37%%41.html`	`i%7a.html` (E)	`i7a.html` (E)

- Preprocessing filters uninteresting mismatches

Performance Evaluation (1)

- Measured overhead for Apache with httperf:
  - 1000 requests to static page
  - 20 connections/second
- Average overhead on the order of 300μs per request



Performance Evaluation (2)

- Impact of overloaded Bro on Apache
  - Outgoing events queued in Apache and eventually dropped
  - Artificially introduced 0.2s delay into Bro's processing
  - No noticeable impact on Apache
- Network load
  - With Nikto's requests, on average 455 bytes/request
    $\Rightarrow$ Scales well with more (busy) Web servers
- Load on Bro
  - Receiving events costs considerably less than parsing HTTP
  - Analyzing additional events is not noticeable

Summary

- Incorporated host-supplied context into a NIDS
  - Context can replace analysis
  - Context can supplement analysis
- Implemented approach for Bro and Apache
  - Apache sends all requests to Bro
  - Bro performs detection and/or comparison
- Installed Apache/Bro in three environments
  - Work well together
  - No performance problems
- Provide client-library to instrument other applications
  - Work-in-progress: Instrumenting SSHD

Enhancing the Accuracy of Network-based Intrusion Detection with Host-based Context

Holger Dreger[1], Christian Kreibich[2],
Vern Paxson[3], **Robin Sommer**[1]

[1]TU München
Germany

[2]University of Cambridge
United Kingdom

[3]ICSI / LBNL
Berkeley, CA, USA

DIMVA 2005